sequence of successive states in a system can look like, or both. Those that are formulated with both concepts use each concept to formalize a different aspect of security [7]. The problem with this approach is that no redundancy is provided for checking that the model accurately captures what we mean by *security*. A security model should be formulated both statically in terms of a secure state and dynamically in terms of a secure transform, and then both formulations should be proven equivalent.

To see that this suggestion is not as preposterous as it may first seem, consider the case of predicate calculus. Besides a definition of a *well-formed formula*, predicate calculus consists of a definition of *provable formula* and a definition of (*semantically*) *valid formula*. The former is provided in terms of a set of proof rules and the latter is provided in terms of a definition of truth in a model, yet both are meant to capture the intuitive notion of *logical truth*. The faith we have in predicate calculus stems from two theorems: the completeness theorem shows that all valid formulae are provable and the soundness theorem shows that all provable formulae are valid. If we take the concepts of *provable formula* and *valid formula* as explicating the concept of *logical truth*, we have two very different definitions of *logical truth* derived by entirely different considerations and formulated in different frameworks that yet, are extensionally equivalent.

This type of redundancy gives us confidence in the correctness of an explication. We may very easily get an explication of an intuitive concept wrong, but to get two explications derived by different considerations wrong in the same way is much less likely. It is such considerations that justify our confidence in *recursiveness* as an explication of *computability*. We are assured by the equivalence of *recursiveness*, λ*-calculability*, and *Turing computability*. Similarly, such redundancy can assure us about the correctness of an explication of *secure system*.

The analogue of this for a security model should be a definition of *security* in terms of a secure state and an alternative definition in terms of a secure transform.[1] We should then prove (1) that if we start in a secure state and apply only secure transforms we will end in a secure state and (2) that if we go from a secure state to a new secure state, our transform must have been secure. These two theorems add support to our model by showing that two distinct formulations of it are identical.

3. The Bell and LaPadula Model

In this section we review the Bell and LaPadula model and its accompanying Basic Security Theorem.[2] The Bell-LaPadula model is based on a state machine in which subjects apply operations (rules) that may require access to objects. The state of the system includes a set of triples that define the current access mode each subject has to each object in the system. Permissible access is determined partly by a security level (classification or clearance) associated with each object and subject. These security levels are partially ordered. Each subject also has a current security level that is bounded above by its clearance. There is also an access matrix that further constrains the access mode an arbitrary subject is allowed to have to an arbitrary object.

The following formal description of the Bell-LaPadula model corresponds to the original notation [3] as closely as possible, but nonessential details are omitted. We assume the existence of the sets S, O, A, and L whose elements are known as *subjects*, *objects*, *access modes*, and *security levels*, respectively. Intuitively, S consists of all system users and programs; O consists of all system files; and A is $\{read, execute, write, append\}$, the set of all

---

[1] For simplicity in comparing formulations, we shall limit ourselves to access control models. Part of the justification for this is, as we shall see, there is an extreme deal of controversy about what the Bell-LaPadula model really is. Trying to attain consensus on what constitutes the proper formulation of an additional existing model for the sake of comparison invites chaos.

[2] Readers already familiar with the Bell-LaPadula model can proceed to Section 4 without loss of continuity.